8T

УДК 004.052.03

Г.Л. Трунов, А.Г. Коваленко
НИИ многопроцессорных вычислительных систем им. акад. А.В. Каляева
Южного федерального университета, г. Таганрог, Россия
GermanLT@mvs.tsure.ru, k.a.g@bk.ru

Параллельно-конвейерная реализация задачи умножения матрицы на поток векторов на реконфигурируемых вычислительных системах

В статье предлагается вариант параллельно-конвейерной организации вычислений при аппаратной реализации решения задачи умножения матрицы на поток векторов на реконфигурируемых вычислительных системах (РВС).

Одной из часто встречающихся операций линейной алгебры является умножение квадратной матрицы на вектор [1]. Входными параметрами для данной процедуры являются: матрица *A[n,n]*, вектор *B[n]*, размерность матрицы *n*. Результирующий вектор – *C[n]*. Каждый элемент результирующего вектора рассчитывается путем перебора элементов строк матрицы *A* и умножения их на соответствующие элементы вектора *B*. Результат умножения прибавляется к текущему значению элемента вектора *C*.

Каждый элемент результирующего массива рассчитывается по следующей формуле:

$$C_i = \sum_{j=1}^{n} A_{i,j} \cdot B_j \, . \qquad (1)$$

Организация параллельных вычислений при решении задачи умножения матрицы на поток векторов на реконфигурируемых вычислительных системах (РВС) позволяет достигать высокого уровня реальной производительности, близкой к пиковой производительности системы [2].

Информационный граф реализации вычислений в соответствии с формулой (1) представлен на рис. 1, на котором указаны входные информационные вершины *A* и *B*, выходная информационная вершина *C*, *m* – глубина линии задержки.

Построение множества вычислительных блоков, реализующих указанный на рис. 1 информационный граф, позволяет производить параллельные вычисления для множества строк матрицы и потока из *m* векторов *B*.

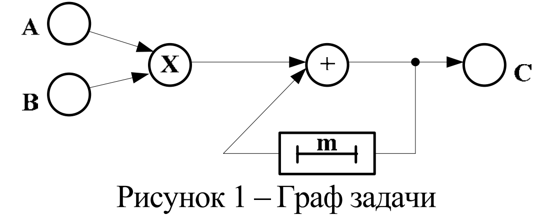


Рисунок 1 – Граф задачи

Таким образом, глубина линии задержки, изображенной на рис. 1, равна величине потока векторов – *m*. При этом поток векторов *B* имеет следующий вид:

$$< b_1^1 b_2^1 b_3^1 .... b_m^1 b_1^2 b_2^2 b_3^2 .... b_m^2 .... b_1^n b_2^n b_3^n .... b_m^n >,$$

где верхний индекс означает элемент вектора, нижний индекс означает номер вектора в потоке.

Таким образом, на входы умножителя с информационных вершин *A* и *B* поступают потоки элементов матрицы и векторов соответственно. Далее, результат умножения складывается с ранее накопленным значением элемента вектора *C*, поступающим из линии задержки.

Для проведения итерации вычислений (умножение одного элемента матрицы на соответствующие элементы векторов) в соответствии с формулой (1) на момент прохождения в информационной вершине *B* порции из *m* *j*-ых элементов векторов в информационной вершине *A* должен быть зафиксирован *j*-й элемент *i*-й строки матрицы.

После прохождения через граф *m* *n*-элементных векторов на выходе сумматора сформируется поток результатов от *i*-й строки матрицы вида:

$$c_{\mathrm{i}} = < c_1^i c_2^i c_3^i \ldots c_m^i > .$$

Таким образом, для организации параллельных вычислений необходимо использовать множество вычислительных блоков, реализующих изображенный на рис. 1 информационный граф, при этом информационная вершина *B* будет общей для всех графов, а на различные информационные вершины *A* подаются элементы различных строк матрицы.

Задача умножения матрицы на поток векторов была реализована на РВС, структурная схема которой приведена на рис. 2. Данная вычислительная структура обрабатывает массивы данных, следующие плотным потоком. Обработка данных ведется в формате с плавающей запятой в строгом соответствии со стандартом IEEE-754.

На рис. 2 изображено решающее поле из 16-ти микросхем Virtex-4 VLX40 с информационными связями между ними, микросхемы распределенной памяти и контроллер обмена КО с управляющим персональным компьютером. Тактовая частота РВС составляет 160 МГц.

Помимо вычислительных блоков структура должна содержать дополнительные блоки, с помощью которых организовывается процесс решения задачи от поступления исходных данных до вывода результатов на данной РВС.

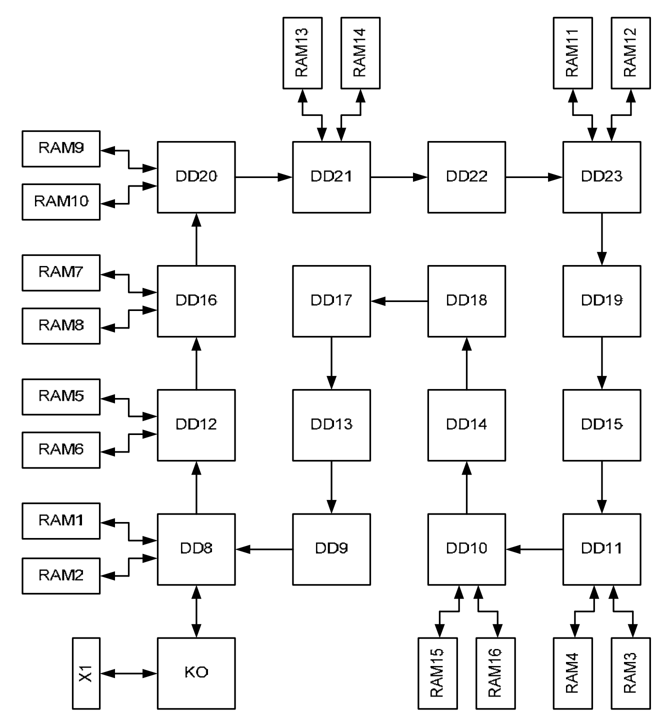


Рисунок 2 – Структурная схема реализации задачи на РВС

Структура вычислителя, предназначенного для решения задачи умножения матрицы на поток векторов, представлена на рис. 3.

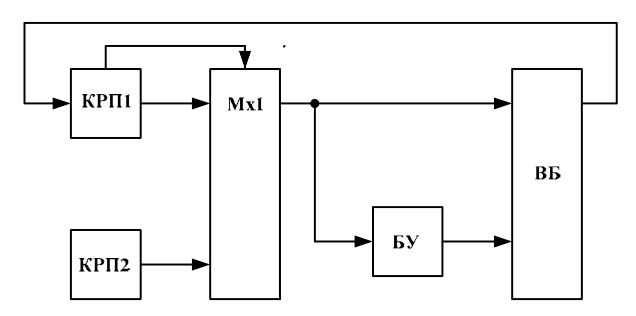


Рисунок 3 – Структура вычислителя

На рис. 3 показаны блоки со следующими обозначениями:
КРП1, КРП2 – контроллеры распределенной памяти;
Мх1 – мультиплексор выбора КРП;
БУ – блок управления;
ВБ – вычислительный блок.

КРП, который описан в [2], обеспечивает связь вычислительной схемы с микросхемами распределенной памяти, из которой считываются команды и данные, а также записываются результаты вычислений.

КРП1 предназначен для хранения и выдачи команд и массива строк матрицы, а также для записи результатов вычислений из ВБ.

КРП2 предназначен для хранения и выдачи массива векторов.

Мх1 предназначен для переключения потоков данных, поступающих на ВБ.

В единый момент времени на ВБ поступает либо массив элементов строки матрицы для сохранения их во внутреннем буфере (загрузка элементов матрицы), либо массив элементов векторов для прохождения через вычислительную структуру (выполнение вычислений). На БУ всегда поступают команды из КРП1.

БУ предназначен для декодирования команд и формирования сигналов, под управлением которых ВБ реализует выполнение задачи.

Структурная схема БУ представлена на рис. 4.

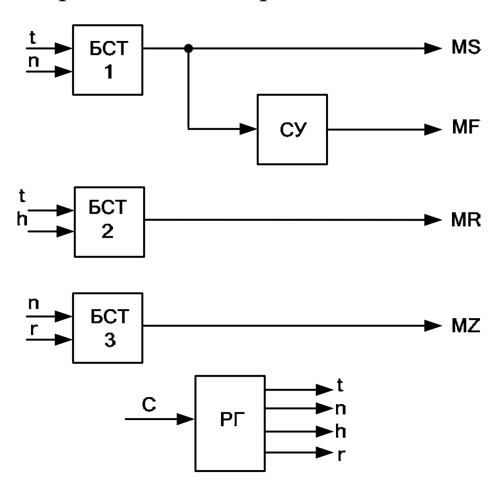


Рисунок 4 – Блок управления

В регистре РГ хранятся параметры вычислительного модуля, поступающие из КРП1 в виде команд *C*:

t – количество векторов, участвующих в обработке;

n – размерность матрицы;

h – количество ВБ в вычислительном модуле;

r – количество загружаемых строк матрицы.

На блоки счетчиков БСТ1 – БСТ3 поступают соответствующие параметры из регистра РГ, что обеспечивает формирование маркеров, сопровождающих различные данные в вычислительном модуле:

MS – маркер для сопровождения потока векторов;

MR – маркер для сопровождения потока результата;

MZ – маркер для сопровождения потока матрицы.

Схема управления СУ формирует маркер MF, сопровождающий поток первых элементов векторов и означающий первую итерацию накопления результатов, при котором на второй вход сумматора в ВБ подается нулевое значение.

Структурная схема ВБ представлена рис. 5.

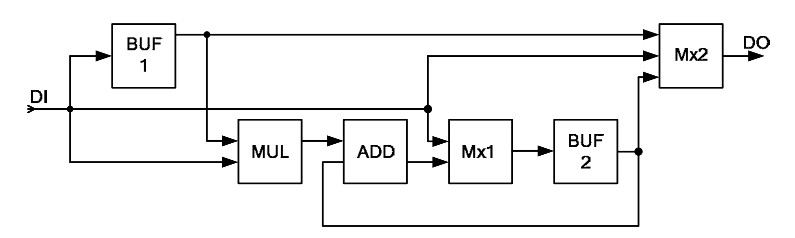


Рисунок 5 – Вычислительный блок

Буфер BUF1 предназначен для хранения строки матрицы. Умножитель MUL, сумматор ADD и линия задержки BUF2 реализуют вычисления в соответствии с информационным графом задачи. Мультиплексор MX2 предназначен для коммутации потоков данных, поступающих в следующий ВБ. Потоки данных представляют собой потоки строк матрицы, потоки векторов и потоки результатов. Мультиплексор MX1 предназначен для коммутации потоков промежуточных результатов и результатов из предыдущего ВБ, поступающих на буфер BUF2.

Таким образом, аппаратная реализация множества ВБ позволяет организовать параллельно-конвейерные вычисления для решения задачи.

Ресурс РВС, представленный на рис. 2, позволяет применить для реализации задачи 8 пар КРП. Также в каждой ПЛИС реализован BUF1 в ВБ, размером $n = 9216$ слов для хранения строки матрицы, и BUF2, размером $t = 1152$ слова для накопления результатов. В микросхемах с КРП размещено по 4 ВБ, а в микросхемах без КРП – по 5 ВБ. Общее количество ВБ в РВС составляет 72, что позволяет вести одновременный расчет результатов по 72-м строкам матрицы.

В соответствии со структурой РВС микросхемы решающего поля разделены на три типа:

DD8, DD12, DD16, DD20, DD21, DD23, DD11, DD10 – с КРП;

DD22, DD19, DD15 – с «транзитным» КРП;

DD14, DD18, DD17, DD13, DD9 – без КРП.

Микросхемы с «транзитным» КРП предназначены для организации синхронизации между КРП. В связи с этим на рис. 6 – 8 изображены измененные структурные схемы вычислений для этих типов микросхем.

Необходимо отметить, что при использовании в вычислительном процессе более одной пары КРП активной (передающей исходные данные и принимающей результаты вычислений) может быть только одна пара. Поэтому в этой структуре использован дополнительный мультиплексор Mx2 для коммутации потоков данных и управления на ВБ от собственного КРП или от предыдущего КРП, когда собственное КРП является пассивным. В связи с этим необходимо ввести процедуру «транзита» (переноса), содержимого

BUF1, в ВБ от текущего КРП к последующему при исчерпании векторов в текущем КРП и от текущего КРП в начальное КРП при исчерпании матрицы в текущем КРП. Количество шагов переноса из КРП в БУ передается также в параметре *r*.

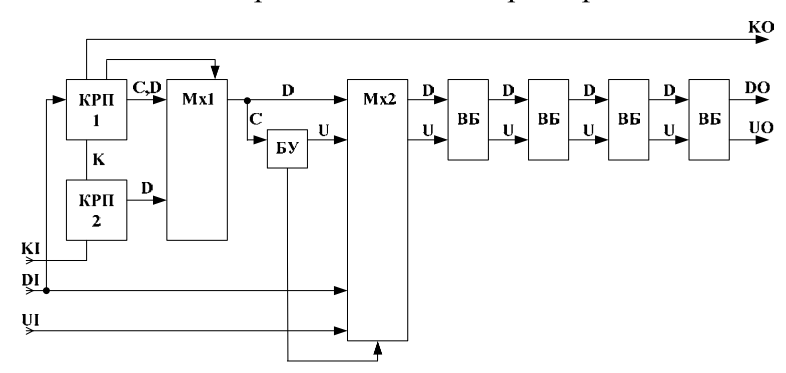


Рисунок 6 – Структура вычислителя для микросхем с КРП

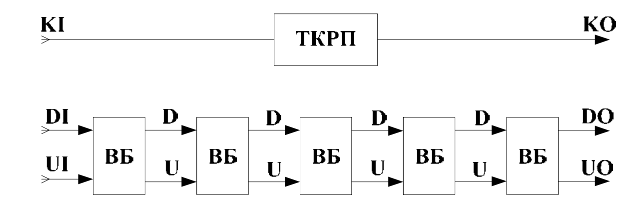


Рисунок 7 – Структура вычислителя для микросхем с «транзитным» КРП

В структуре, показанной на рис. 7, применен блок «транзитного» КРП (ТКРП) для синхронизации всех КРП БМ. Отсутствие мультиплексоров и БУ объясняется отсутствием КРП.

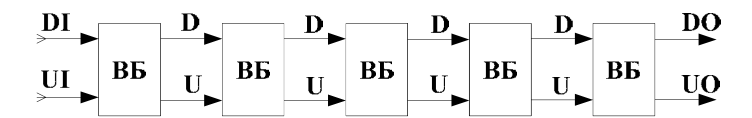


Рисунок 8 – Структура вычислителя для микросхем без КРП

В последней структуре (рис. 8) использованы только ВБ.

Обобщенный алгоритм функционирования вычислительного модуля состоит из трех операций, выполняемых последовательно во времени:
– загрузка строк матрицы;
– чтение потока векторов (выполнение вычислений);
– сохранение результатов.

Таким образом, при реализации на данной РВС рассмотренная структура вычислителя способна решать задачу умножения матрицы на поток векторов со следующими параметрами:

максимальный размер матрицы – 9216
максимальный поток векторов – 14560

максимальная размерность матрицы ограничена объемом BUF1 в ВБ, а максимальное число векторов определяется объемом памяти, подключенной к КРП.

За один шаг структура может обработать:
векторов – 1152
строк матрицы – 72.

Следует отметить, что решить задачу с обоими максимальными параметрами (размер матрицы и число векторов) не представляется возможным. При решении задачи с максимальным значением размера матрицы допустимая величина потока векторов будет меньше приведенного и наоборот. Это связано с тем, что в процессе решения задачи накапливается массив результирующих данных, размер которого, в итоге, превышает свободный объем памяти, хранящей исходные строки матрицы.

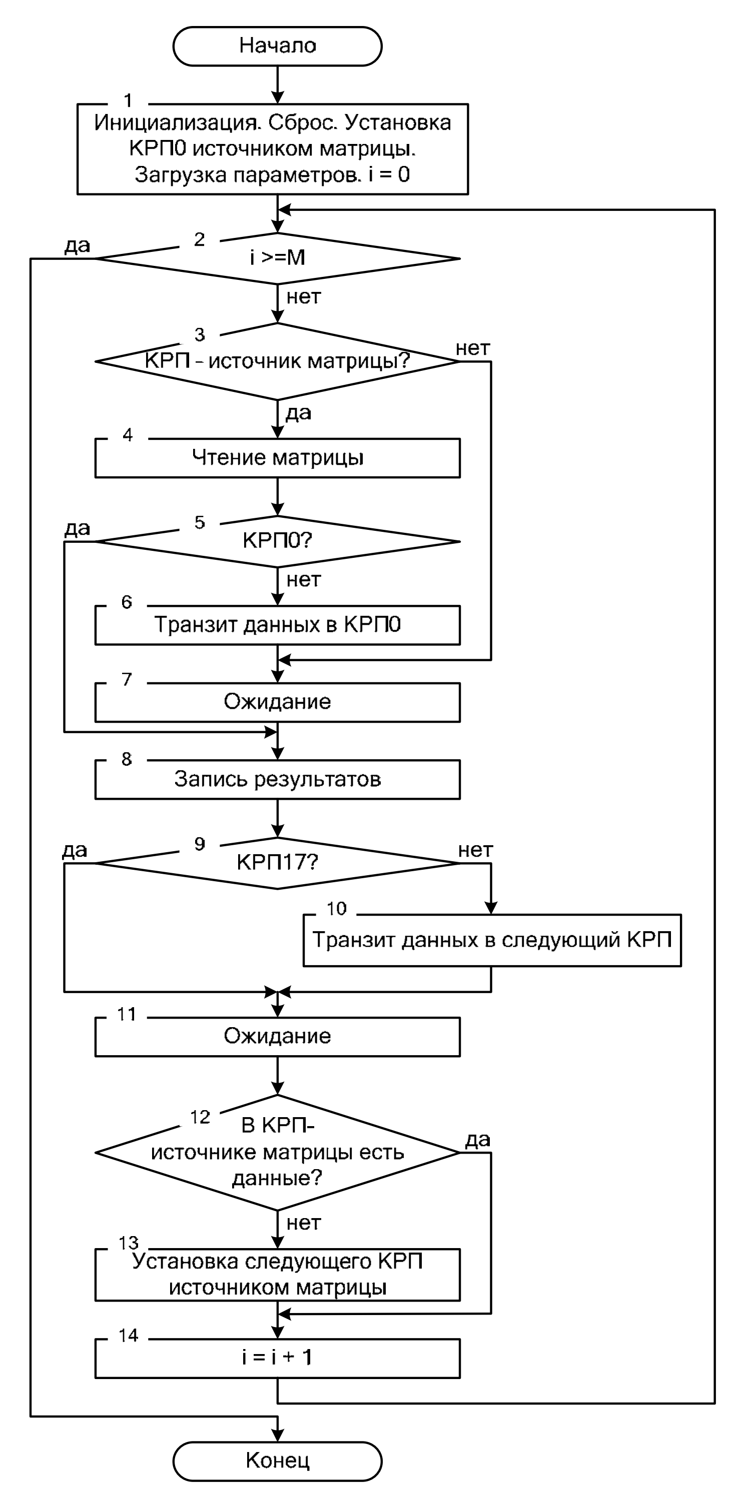


Рисунок 9 – Блок-схема алгоритма работы ведущих КРП

Для управления работой вычислительной схемы была разработана параллельная программа на языке Argus. Данная программа представляет собой набор инструкций (команд) для КРП, осуществляющих:
– чтение или запись данных в схему или из схемы;
– переключение режимов работы КРП;
– последовательную работу всех пар КРП, когда обрабатываются строки матрицы и векторы в одной паре КРП, другие КРП находятся в режиме ожидания;
– изменение таких параметров, как КРП-источник данных, число пройденных итераций;
– проверку необходимости перехода к следующей итерации.

В каждой паре КРП одно КРП является ведущим, а другое – ведомым, и, следовательно, для каждого из них существует собственная подпрограмма.

Блок-схема алгоритма работы ведущих КРП показана на рис. 9.

На этом рисунке в блоке 1 происходит инициализация переменных начальными значениями, подается программный сброс в вычислительную схему, загружаются необходимые параметры.

Далее организуется цикл по i (блок 2), который повторяется до тех пор, пока не выполнится условие $i>=M$, где M – параметр, определяющий общее количество блоков исходной матрицы по 72 строки в каждом блоке. Число строк в блоке обусловлено характеристиками вычислительной схемы.

В блоке 3 проверяется условие, по которому КРП либо читает строки матрицы в вычислительную схему (блок 4) с последующей проверкой (блок 5) о необходимости транзита прочитанных строк в КРП0 (блок 6), либо переходит в режим ожидания (блок 7). Таким образом, на этом участке алгоритма учитывается тот факт, что для КРП0 нет необходимости производить транзит данных, потому что данные и так находятся в КРП0.

В блоке 8 происходит запись полученных результатов.

Если текущий КРП не является последним (номер последнего ведущего КРП – 17) (блок 9), то осуществляется транзит строк матрицы в следующее КРП (блок 10). После этого текущий КРП переходит в режим ожидания (блок 11), пока остальные КРП не закончат свою работу по записи результатов. Следует отметить, что режим ожидания означает перевод текущего КРП в пассивное состояние.

В конце алгоритма для текущего КРП-источника данных проверяется условие наличия строк исходной матрицы (блок 12). Если условие не выполнилось, то источником данных устанавливается следующий КРП (блок 13).

После наращивания параметра цикла (блок 14) происходит переход на следующую итерацию.

Блок-схема алгоритма работы ведомых КРП показана на рис. 10.

Основной задачей ведомого КРП является выдача потоков векторов в заранее определенные моменты времени под управлением ведущего КРП.

Для простоты реализации принято, что:
– исходная матрица и поток векторов распределяются на ресурс оперативной памяти РВС равными долями;
– максимальное число векторов составляет 9216, равное максимальной размерности матрицы.

Схема БУ, ВБ и КРП реализована в САПР ISE 7.1. Проведено функционально-временное моделирование проекта в программной среде Model Sim 6.1., которое показало правильность выполняемых вычислений, а также возможность работы БУ и ВБ на частоте 250 МГц.

Также задача умножения матрицы на поток векторов была реализована на РВС другой архитектуры, содержащей 16 ПЛИС фирмы Xilinx XC4VLX80 с тактовой частотой 250 МГц, в которых размещено 12 пар КРП.

Экспериментальные исследования параллельно-конвейерной реализации задачи умножения матрицы на поток векторов показали, что производительность РВС, содержащей 16 ПЛИС фирмы Xilinx XC4VLX40 с тактовой частотой 160 МГц, составила 21,7 ГФлопс, на РВС, содержащей 16 ПЛИС фирмы Xilinx XC4VLX80 с тактовой частотой 250 МГц, – 70,9 ГФлопс.

Выигрыш по времени решения по сравнению с ПК Intel(R) Core(TM) 2 CPU4400 @ 2.00GHz составил 54 раза и 176 раз соответственно.

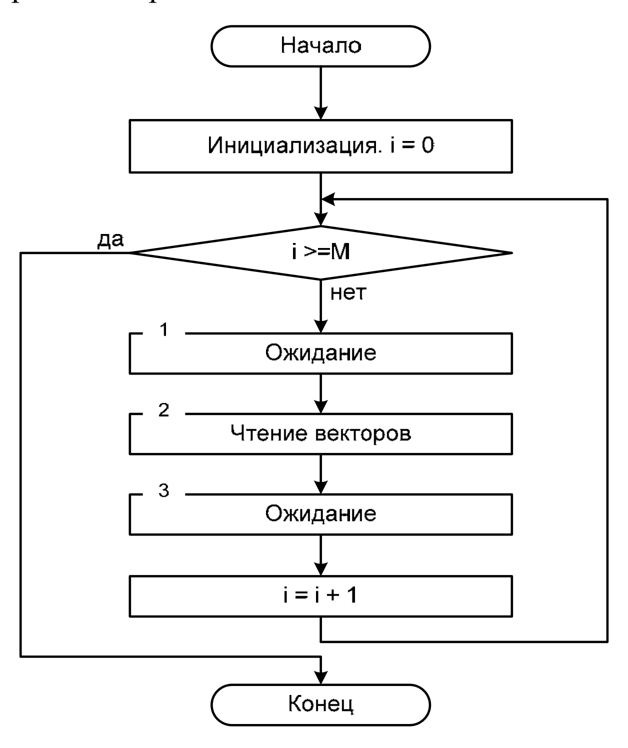


Рисунок 10 – Блок-схема алгоритма работы ведомых КРП

Таким образом, предложенная организация параллельно-конвейерных вычислений при решении задачи умножения матрицы на поток векторов позволяет достигать высокой реальной производительности реконфигурируемой системы, существенно превышающей производительность современных универсальных процессоров.

Литература

1. Уоткинс Д. Основы матричных вычислений. – М.: БИНОМ, 2006. – 664 с.
2. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Янус-К, 2003. – 380 с.

Г.Л. Трунов, О.В. Коваленко

Паралельно-конвейєрна реалізація задачі множення матриці на потік векторів на обчислювальних системах, що реконфігуруються

У статті пропонується варіант паралельно-конвейєрної організації обчислень при апаратній реалізації рішення задачі множення матриці на потік векторів на обчислювальних системах, що реконфігуруються.

Статья поступила в редакцию 21.07.2008.